Routing in Sensor Networks: Performance and Security in clustered networks

M. A. Abuhelaleh, K. M. Elleithy
School of Engineering, University of Bridgeport
Bridgeport, CT 06606
{mabuhela, elleithy} @bridgeport.edu

Abstract-

Due to high restrictions in sensor network, where the resources are limited, clustering protocols for routing organization have been proposed in much research for increasing system throughput, decreasing system delay and saving energy. Even these algorithms have proposed some levels of security, but because of their dynamic nature of communication, most of their security solutions are not suitable. In this paper we focus on how to apply the highest possible level of security to sensor networks and at the same time increase the performance of these networks by changing the way that sensors communicate with each other.

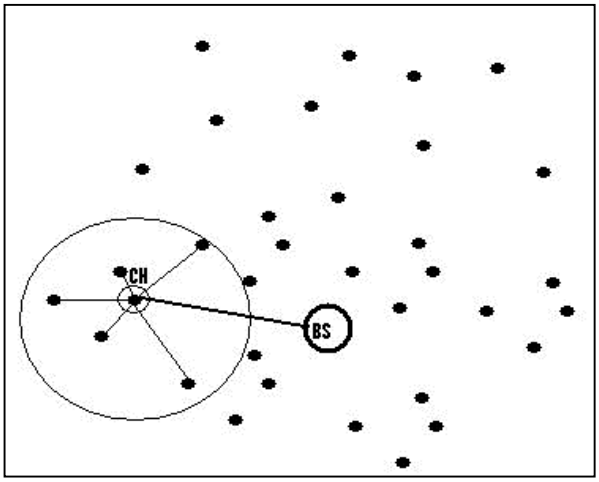

Figure1. Cluster organization for sensor networks

I. INTRODUCTION

There are many advantages of using sensor networks. They provide dynamic and wireless communication between nodes in a network, which provides more flexible communications. At the same time, sensor networks have some special characteristics compared to traditional networks, which makes it hard to deal with this kind of networks. The most important property that affects this type of network is the limitation of the resources available, especially the energy.

Sensor networks are self organized networks, which makes them suitable for dangerous and harmful situations. At the same time makes them easy targets for attack. It is important to apply some level of security so that it will be difficult to be attacked, especially when they are used in critical applications [1].

Wireless Sensor Networks (WSNs) [2] are special kinds of Ad hoc networks that became one of the most interesting areas for researchers. Routing techniques are the most important issue for kind of network where resources are limited. Cluster-based organization has been proposed to provide an efficient way to save energy during communication [3]. In this kind of organization, nodes are organized into clusters. Cluster heads (CHs) pass messages between groups of nodes (group for each CH) and the base station (BS), (Figure1). This organization provides some energy saving which is the main advantage for proposing this organization. Depending on this organization, LEACH (Low Energy Adaptive Clustering Hierarchy) [3] enhanced security, where the CHs are rotating from node to node in the network making it harder for intruders to know the routing elements and attack them. [4]

There are some existing works to improve the security of LEACH. Recent techniques provide efficient security to pairwise node-to-CH communication [5]. In [5], a modified version of LEACH that inherits its security from random key distribution was proposed.

In this paper, we discuss some existing work of LEACH and we focus on two important criteria; the performance and the security. In section two, we discuss the original work of LEACH, and then in the third section we discuss two of the most interesting modifications proposed for LEACH to increase performance and security. In the fourth section we discuss the security and performance of LEACH, and propose modification to increase the performance, and to improve the security. In section 5, we evaluate performance and security of our solution compared to other solutions.

2. LEACH protocol

LEACH was first proposed to reduce total energy consumption in sensor networks [3]. It is assumed that every node can directly communicate with a BS using a high enough transmitting power. By providing a clustered hierarchy, we can balance the energy consumption. Sensor nodes send their messages to specific nodes and they will be considered as cluster heads (CHs). CHs then aggregate these messages and send them to the BS. We can notice that this process results in energy saving for nodes that are not involved in CHs since they can transmit now with less transmission power, but at the same time we consume the energy of CHs. To solve this problem, LEACH proposed a dynamic CH rotation which concluded that the CHs should change at each round. Every round, a new node will become a CH. The network chooses CHs using a distributed algorithm and then dynamically clustering the remaining nodes around CHs.

2.1 Description

LEACH is working in rounds. We will summarize the steps for a single round in the remaining part of this section.

According to oliveria *et.al*, each round consists of two main phases; the setup phase (initial phase) and the steady state phase (real transmission phase) [5]. For the setup phase, each node decides the probability that it can be a CH for the current round, considering the energy and the knowledge of the desired percentage of CHs. Let us call these Ready Nodes RNs. RNs broadcast advertising messages for the whole network. When the nodes receive all advertising messages, the remaining nodes will choose a CH depending on the highest signals received from RNs and then each of these nodes will send a message to the desired CH requesting to join it. When the CHs receive the messages, they start to broadcast the confirmation for these accepted nodes by sending confirmation messages with a time slot schedule for each node in the group. This time slot tells each node in that group when it is time to transmit its messages.

The second phase concerns the transmission of the real data among the network. According to the time schedule provided by CHs to other nodes, each will start sending its data to the proper CHs. CHs then will collect the messages from their members, analyze and handle them, then send the results to the BS.

2.2 Security in LEACH

Jamming and spoofing are kinds of attacks that could be harmful to sensor networks. In LEACH, the nature of clustered distribution can lead to a harmful attack, especially when that attack relies on CHs for sending and receiving data. If a hacker decides to become a CH, this can result in a disrupted network. Selective forwarding and sinkhole attacks are examples of these kinds of attacks [5].

In LEACH, the possibility for the network to be attacked by these kinds of attacks is very small because CHs are changing in each round of communication. It is hard for the intruders to know the expected CHs for each round so that they can disrupt the critical points of the network.

3. Improving LEACH

By analyzing the work of LEACH we can determine the critical points of communication, and then we can focus on providing more efficient security at those points. One point is to determine CHs in a way that it will be hard for the intruder to guess which nodes will be CHs.

The easiest and the most efficient way is to prevent suspicious nodes from participating in the network, and this step should be taken at the time of network setup.

By providing a secure way to prevent illegitimate nodes from participating in the network we can reach a good level of security and we can reduce the future work load of the network to provide security. Some studies propose controlling access to the network for sensor networks, and most of these works are based on key distribution (KD) for cryptographic mechanisms.

3.1 Existing work on LEACH

As we mentioned earlier, there are many techniques proposed as new modifications for LEACH to provide more security and to reduce energy consumption. In this section we will discuss two of these works and then we will propose some modifications for these works.

3.1.1 F_LEACH

F-LEACH [6] is an enhanced version of LEACH that gives protection for the network. It suggests that each node has to have two symmetric keys: a pairwise key shared with the BS and the last key chain held by the BS. According to that, it suggested small modifications for LEACH. For the setup phase, the message sent by RNs should consist of an encrypted message that contains the ID of the node that should receive the message and the ID of CH itself as plain text, and the encryption (ID of CH, the counter shred by CH and the BS, and the advertisement message) using the message authentication code (MAC) that is produced using the shared key between CH and the BS.

The nodes hold the ID's of the CHs. At the same time the BS will analyze the messages sent by CHs to authorize them. Any valid CH will then have its ID added to the list of valid nodes IDs. After that, the BS broadcasts the list with the encrypted list for all nodes in the network using $\mu TESLA$ [7] broadcast authentication scheme. Now the nodes can recognize the authenticated RNs to be connected with, so these nodes send their requests to participate with CHs groups. CHs then broadcasts confirmation messages for approved nodes. Each message will contain the time slot schedule for each node.

It can be noticed that F_LEACH does not provide full authentication for node-CH where the messages to be sent from the nodes to CH are not authenticated.

Oliveria *et.al* proposed another solution to provide some ways to redistribute the keys using random key redistribution for securing node-CH communication in LEACH [5].

3.1.2 SecLEACH- Random KD to LEACH

SecLEACH [5] proposes improvements to LEACH protocol. It shows how to invest the key redistribution scheme to secure node-to-CH communications. The main idea is to generate a large pool of

keys and their IDs at the time the network is deployed, and then each node is assigned a group of these keys randomly. Also, each node is assigned a pairwise key which it shares with the BS.

In setup phase, each CH includes the IDs of the keys in its key group, and a nonce in its advertising message offering its availability to become a CH. The ordinary nodes also choose an ID that is shared with CH. Then each of these ordinary nodes sends the message to CH requesting to join its group.

The message includes the ID of the node, ID of CH, r, join_ request message, and the encryption (node ID, CH ID, r and the nonce sent by CH) using MAC that is produced using a symmetric key associated with r. Each CH then sends a confirmation message to approved nodes containing the ID of CH and a group of pairs (ID and time slot for each node to start transmission).

In steady state phase, the nodes transmit the messages to CHs according to the time slot provided before, and each message includes the ID of the node, the ID of desired CH, sensing report from the node, and the encryption (node ID, CH ID, node sensing report and the nonce+ reporting cycle within the current node) using the same MAC used before. Finally, CH starts sending the final data for the BS, and the message includes the ID of CH, the ID of BS, the aggregation data from all nodes, and the encryption (the aggregation data and the ID of CH) using the MAC produced from the ID of CH.

This algorithm provides authenticity, confidentiality, and freshness for node-to-node communication. The security level is not impacted by the number of nodes; actually it depends on the size of the key group assigned for each node according to the total size of the key pool [5].

4. ModLEACH – Performance and Security

In this section we first discuss the main weakness points for the previous work according to the performance and security, then we show how to improve the performance, and finally we propose some modifications to security proposed by LEACH.

4.1 Security and Performance in LEACH

F_LEATCH does not manage to provide a complete and efficient solution for node-to-CH authentication [6]. In SecLEACH, this problem has been solved in an efficient way, but most of the work has to be controlled by the BS which will cost more than what is expected according to [8]. Data overload has also increased compared to original LEACH.

LEACH itself is a smart way of work organization in sensor networks, but the problem here is how to determine the elected CH each round in the network. As we can see from previous description of LEACH, most of these decisions have to be made by the BS.

4.2 Improving Performance

We can improve the performance of LEACH and its related works by changing the way to elect CH. We propose a simple way to rotate the CHs from round to round.

Instead of building the clusters directly by sending an advertised message and receiving a join request message, the nodes start broadcasting directly to direct neighbors, then a node with high enough power broadcasts an acknowledgment to desired nodes with their time slot schedule. The first acknowledgment received indicates the first possible CH, then after the node checks the validity of this CH using some security techniques, it elects this node as its CH, and it discards any other messages received from other nodes.

Routing to the most direct neighbors decreases the energy required for transmission, since the distance becomes less. Also the number of messages needed to build the clustered network is less. The node, just for each round, sends its first real message to the next neighbors. After receiving the acknowledgment from the first RN, it will consider this node as CH and start to broadcast the rest of the messages through this CH to the BS. Then, by default, after each round the energy in these CHs will be less than before, so they will not be ready nodes in most cases. New nodes will take the leadership after each round.

4.3 Improving Security

Security is handled very well in SecLEACH. In our protocol we use the same keys that were proposed by SecLEACH as follows: SecLEACH suggests generating a large pool of keys and their IDs due to the deployment of the network. Each node has to be assigned by a group of keys (m) by using a pseudorandom function to provide the node ID which will be used to generate a sequence of keys ID that will be assigned to that node. Also at the time of network deployment, each node has to be assigned by a pairwise key shared with B.S. (this key will be used for verifying the aggregate messages sent by CHs).

4.4 ModLEACH protocol

The main modification for LEACH is to change the way that the CHs are elected and to change in advance the way that the nodes and CHs communicate.

After generating the key pool and assigning the groups of keys for each node, in addition to a pairwise key for each of them, we can start our protocol. First, each node that needs to send data to base station starts sending the first message from its sequence of messages. The main idea here is to try join the nearest nodes considering them as possible CHs for the current round (by this step we can skip the setup phase proposed by LEACH and other related protocols). The nodes broadcast their first packet. The packet includes necessary information: TTL (Time to Live), node ID, ID of common key used in encryption (CK ID), the tag number TG=1 (1-bit number / 0 for join request, 1 for approval), the status number ST=0 (1-bit number /0 for first packet send, 1 otherwise). All this information is encrypted using MAC that produced using the common key mentioned before. Also the packet contains the plain text of node ID, the tag number, and the status number. Next the node will wait for enough time to see if there is any response. TTL is set to one so that only the direct neighbors will receive the packet [9]. All direct neighbors will receive the packet. Each node checks the TG and ST numbers; if TG is zero and ST is zero then that means the node is requesting from the reception to become its CH. If the node has enough power then it can reply to the nodes that sent the packets to join the group of CH. Then for each packet RN receives, it will store the node ID in a possible member table with a sign that it is not approved yet and sends a confirmation packet to all accepted nodes that includes: TTL, TG=1, ST=0, RN ID, and node IDs all encrypted using MAC that is produced using the common key for CH which is shared between it and the desired node. According to Eschenauer and Gilgor, any two sensors can be assured to have one sharing key at least if the number of common keys is reasonable [10]. In addition to that information, the RN ID, CK ID, and the tag are also included in the packet as a plain text. When the desired node receives the confirmation packet, it will check as before for the tag number; if it's zero, then it is a request to become CH. In this case the previous step will be repeated. Otherwise it is a confirmation; in this case it will process the packet to check the validity of the CH. If it is valid, it will start sending the rest of the packets for the elected CH. The parameters for next packets are like before, with the sensing report, TG=0, ST=1. When CH receives the message with those parameters values, it will be as an approval for it to become a CH, where at least one node has to approve RN to be a CH, and it will change the status of the node in its member table to be approved. So it will start collecting the packets from each node in member table. Then it will aggregate these data in one message and start sending it to the BS. Each packet sent from CHs to the BS includes: the ID of CH, the ID of B.S, the aggregation of data, and the

encryption (data aggregation and CH ID) using the MAC address using the associated symmetric key associated with CH ID. This one is shared with the BS which allows the BS to verify the validity of the CH and the message.

In this scenario, a case of no available direct neighbor to be a CH is possible but this possibility will be small, since each node send to its direct neighbor. If this case happens, then there are two available solutions: first, the node can retransmit the first packet with increasing the value of TTL to two. This will produce wider range of direct neighbors than before. The second solution is the rest of nodes that don't have CH to follow, may send directly to the BS. We preferred the second solution, because the first solution has another possibility: at the time the nodes send another request, all CHs can be reserved. Another reason is it that will increase the risk that some intruders can be involved in this request, which at least will result in delay of transmission.

5. Evaluation and Analysis

Since we built our solution on the same ideas followed by SecLEACH, which is mainly using KD, we will start for each part of this section by showing the results provided by [5, 10]. We will provide the mathematical model that shows improvement offered by our solution.

5.1 Performance Evaluation

In Wireless Sensor Network (WSN), there is a fixed space for each node to store the key group selected from the key pool. This means that the size of the group (GS) is fixed at the first time the network is built. After GS is determined, the size of the key pool (PS) will affect the network in two ways

1. Level of security:
   Depending on the variable names provided before, the security level is given by [5]

   Security level= 1- GS/PS

   This means: increasing the PS will provide us with higher level of security.

2. Sharing keys probability:
   The probability of two nodes not to share the key is given by the formula [5]

   $P= [(PS-GS)!]^2 / PS!*(PS-2GS)!$

This means that the probability for two nodes to share the key is increased by increasing the size of the key pool.

Since we used the same technique to generate the key pool and to provide the key groups, then the issue of key sharing technique will get the same performance proposed by SecLEACH.

Because all CHs use the same single hop to communicate with the B.S, then increasing the number of CH will lead to more power consumption. In our solution we followed the KD scheme used by SecLEACH to produce the sharing keys, and as we mentioned before, increasing the size of the pool will decrease the number of CH produced, where only the nodes that received the first packet and share the same key can then proceed with the communication. On the other hand, decreasing the number of CH may results in increasing the number of nodes that joined the CHs.

Providing a suitable size of key pool leads to suitable level of security with high performance, see (Figure2).

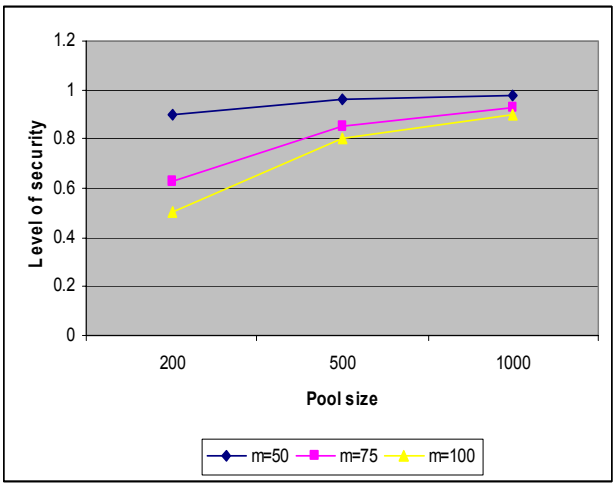

Figure2. Security level affected by key pool size and the keys group size, m represents the size of each group.

Overload and number of messages needed to provide a complete round are not taken as important issues in LEACH, SecLEACH and FLEACH, even as it approved in the study by Eschenauer and Gligor that in general, the cost of data transmission is more than data processing [10], and since the overload of the packets could affect the transmission in order to waste more energy, we proposed our solution.

Our solution proposes one phase that combines the setup and steady phases together. First the node sends the requesting packet to the RN with some extra bits, which are even less than the real packet sent from the node to CH in pervious LEACH protocols. Then the RN will check the variables in the packet to determine the type of this packet. According to its type it responds by replying to the nodes and storing the ID of the node as an expected member at the same time. Then RNs send messages to indicate that they are ready to be CHs with time slots for each node. Nodes then reply to CHs by the real message including the sensing report, then CHs will send the aggregated data from their member, and it will send it to the proper BS.

5.2 Security evaluation

As it discussed in [5], ModLEACH that follows the same techniques that proposed by SecLEACH for security. The level of security to prevent the node capture by intruders is determined by the security level of the network. Using KD scheme, the size of the key pool determines the level of security provided by my solution. Increasing the size of the key pool increases the level of security.

6. Conclusions

The advantage of using clustered organization is the total energy consumption. LEACH is one of the best schemes that applied this kind of organization, where it provides a dynamic rotating of clusters heads at each round. Some security solutions have been proposed for LEACH to improve the security level in such scheme. The overload is not considered carefully in these improvements. In our proposed we decrease the overload, improve security, and decrease the number of transmissions needed to complete the communication.

Biographical Information

Mohammed Abuhelaleh is a full-time Ph.D. student of Computer Science and Engineering at the University of Bridgeport. He worked as a lecturer for some computer science courses in addition to college courses like Data structure, computer skills 1 & 2 in Alhusein Bin Talal University / Jordan for three years. He has master degree computer science from University of Bridgeport, and graduated with a GPA of 3.48. Mohammed now is in second semester of PHD program, and he is working as a graduate assistant for prof. Elleithy at Engineering and Computer Science department at the University of Bridgeport.

Dr. Elleithy received the B.Sc. degree in computer science and automatic control from Alexandria University in 1983, the MS Degree in computer networks from the same university in 1986, and the MS and Ph.D. degrees in computer science from The Center for Advanced Computer Studies at the University of Louisiana at Lafayette in 1988 and 1990, respectively. From 1983 to 1986, he was with the Computer Science Department, Alexandria University, Egypt, as a lecturer. From September 1990 to May 1995 he worked as an assistant professor at the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. From May 1995 to December 2000, he has worked as an Associate Professor in the same department. In January 2000, Dr. Elleithy has joined the Department of Computer Science and Engineering in University of Bridgeport as an associate professor. In May 2003 Dr. Elleithy was promoted to full professor. In March 2006, Professor Elleithy was appointed Associate Dean for Graduate Programs in the School of Engineering at the University of Bridgeport. Dr. Elleithy published more than 100 papers in international journals and conferences. He has research interests are in the areas of computer networks, network security, mobile communications, and formal approaches for design and verification.

REFERENCES

[1] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. IEEE Computer, 35(10):54–62, Oct. 2002.

[2] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In Mobile Computing and Networking, pages 263–270, Seattle, WA USA, 1999.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In IEEE Hawaii Int. Conf. on System Sciences, pages 4–7, january 2000.

[4] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. Elsevier's Ad- Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols, 1(2–3):293–315, 2003. Also appeared in 1st IEEE International Workshop on Sensor Network Protocols and Applications.

[5] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, A. A. F. Loureiro. SecLEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks. Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)

[6] A. C. Ferreira, M. A. Vilac¸a, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. F. Loureiro. On the security of cluster-based communication protocols for wireless sensor networks. In 4th IEEE International Conference on Networking (ICN'05), volume 3420 of Lecture Notes in Computer Science, pages 449–458, Reunion Island, April 2005.

[7] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. Wireless Networks, 8(5):521–534, Sept. 2002. Also appeared in MobiCom'01.

[8] Kun-Won Jang, Sang-Hun Lee, Moon-Seog Jun. Design of Secure Dynamic Clustering Algorithm using SNEP and μTESLA in Sensor network. 2006 International Conference on Hybrid Information Technology - Vol2 (ICHIT'06)

[9] Fen-hua Cheng, Jin Zhang, Zheng Ma. Curve-Based Secure Routing Algorithm for Sensor Network. 2006 International Conference on Intelligent Information Hiding and Multimedia.

[10] Eschenauer and V.D.Gligor, "A key-management scheme for distributed sensor networks", Proc. of the 9th ACM Conf. On Computer and Communications Security, pp.41-47, 2002.